CS 3710 Introduction to Cybersecurity
Term: Fall 2025

Lab Exercise 7 – Binary Analysis, Firewall, and Intrusion Detection

Due Date: November 21, 2025 11:59pm
Points Possible: 7

Name:

AI assistance is permitted on this assignment. Please be aware that AI answers are not always correct, so validate the answer. If you are opposed to using AI, you do not need to do so on the questions specifically requesting AI. Please cite all sources including AI.

1. Overview

This lab exercise will provide some hands-on experience with binary analysis, firewall configuration, and intrusion detection.

2. Resources required

This exercise requires Kali Linux VM running in the Virginia Cyber Range.

3. Initial Setup

From your Virginia Cyber Range course, select the **Ubuntu with Snort and Other Tools** environment. Click "start" to start your environment and "join" to get to your Linux desktop login. This environment requires authentication. Log in using these credentials:

Username: **`student`**
Password: **`student`**

Once you are logged in, click the Terminal Emulator in the bottom menu to open the command line.

4. Tasks

Task 1: Fuzzing

Run the file **game** and win the game by capturing the flag. You can use the various static and dynamic tools and some fuzzing to determine the best way to trick the game and win. The file is already located on the Desktop in Cyber Range but you may need to install the analysis tools you want to use. If you want to use your own system you can download the game file here:

bit.ly/3Cc5QEo

Question 1: **Provide a screenshot of the flag and explain what you did to win the game.** **(.5 point)**

```
student@ip-10-1-69-204:~/Desktop$ ./game
Beat the house and win! Can you get to $1M without going bust?
You begin with $1K
Your pot is 1000

Enter integer bet: -100000000
The house rolls 6

Press [enter] to roll
You roll 5
Bad luck. You lose.
You won!
Access Granted!
flag:theresaneasywayandahardway.
student@ip-10-1-69-204:~/Desktop$
```

Task 2: Firewall Configuration

[Note: be very careful with firewall rule configuration changes on your Cyber Range virtual machine. If you set the rules improperly you could break your network connection to the range VM. Fortunately, this can almost always be fixed by restarting your VM. If that happens, go to the Virginia Cyber Range page and select the "Stop Exercise" button for this lab, then restart the exercise and re-join.]

Use the following command to set the host-based firewall on your Linux system to a default policy that we have specified:

```
$ sudo /etc/default_firewall.sh
```

[When using ***sudo*** you may need to enter your student password: ***student***]

Linux host-based firewalls are configured using the **iptables** command. There is a pretty good (and short) tutorial at http://fideloper.com/iptables-tutorial. To review the firewall rules set by the default policy, use the following command (you may want to use the mouse to drag the terminal screen wider first to read the output more clearly with no linebreaks):

```
$ sudo iptables -L -n
```

Simple packet filtering firewalls usually have a default policy to DROP (deny) packets and only to accept traffic that meets specific criteria. When a packet arrives on a host, the firewall tries to match firewall rules starting with the first rule in the chain. The firewall will apply the first rule that matches and the default rule is applied last, so if there is a rule that "ACCEPTs" a packet before the default DROP, the packet will be accepted. In general, if a specific input or output IP address, port, or protocol is not specified, the rule applies to 'any' IP address, port, or protocol.

Review the default firewall configuration (**`$ sudo iptables -L -n`**) and answer questions 2 – 5.

Question 2: **What is the default policy on the INPUT, FORWARD, and OUTPUT chains in the default firewall configuration and what does this mean for each? (.5 point)**

INPUT chain: default policy is DROP. This means that any incoming packet to this host that doesn't match an earlier ACCEPT rule will be blocked.

FORWARD chain: default policy is ACCEPT. This means packets that are being forwarded through this host are allowed by default (unless a specific rule later drops them).

OUTPUT chain: default policy is ACCEPT. This means packets leaving this host are allowed by default unless some more specific rule blocks them.

Question 3: **What specific firewall rules are in place on the INPUT chain? Specify protocols and ports for which packets are allowed by the rules provided, and under what conditions those packets are allowed. (Hint: there are 5 rules) (.5 point)**

1. **ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0**
2. **ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0**
3. **ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:22**
4. **ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW,ESTABLISHED tcp dpt:33**
5. **ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state NEW,ESTABLISHED**

Question 4: **You notice a big problem with the firewall rules on the INPUT chain. What is it? (.5 point)**

The "big problem" is the last rule on the INPUT chain:
ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state NEW,ESTABLISHED

This rule accepts all new incoming connections on any port from any IP as well as established ones. Since it comes before the default policy drop, it overrides the whole firewall and essentially means the host is open to any inbound traffic instead of restricting it to specific ports.

Question 5: **What firewall rules are in place on the OUTPUT chain? Specify protocols and ports for which packets are allowed by the rules provided, and under what conditions those packets are allowed. (.5 point)**

There are three rules:

- TCP, port 22 (SSH) ACCEPT tcp ... state ESTABLISHED tcp dpt:22
- TCP, port 33 ACCEPT tcp ... state ESTABLISHED tcp dpt:33
- All protocols/ports ACCEPT all ... 0.0.0.0/0 0.0.0.0/0

We will use two shell scripts to modify the firewall configuration. A script called '/etc/extingui.sh' will clear all firewall rules and set the default policy on the INPUT, OUTPUT, and FORWARD chains to ALLOW all traffic in and out of your server. Execute this script as follows.

```
$ sudo /etc/extingui.sh
```

Perform the following command again to see that the firewall rules are cleared:

```
$ sudo iptables -L -n
```

Once the firewall rules are cleared, you will modify the script '/home/student/lab2/firewall.sh' to add firewall configuration commands. This file is not a blank file, it already has a firewall rules template in it. Use the text editor of your choice to edit this script (one option is “mousepad”.)

```
$ mousepad /home/student/lab2/firewall.sh
```

Iptables commands are of the following form:

```
iptables [command-type] [pattern-match options] -j [action]
```

Where `[command-type]` specifies whether the rule will be added or deleted on a specified chain, `[pattern-match-options]` specifies the port, interface, address, etc. to match, and `[action]` specifies what action to take if the packet matches the pattern (DROP, REJECT, ACCEPT, LOG).

In our simple packet filtering firewall, all of our rules will be added to the INPUT or OUTPUT chains and our actions will either be ACCEPT or DROP; so in this exercise, all of your rules will be of the form:

```
iptables -A INPUT  [pattern-match options]  -j [ACCEPT or DROP]
```

Pattern match options that you will use include:

-s	source IP address or address range (can use CIDR addressing)
-d	destination IP address or address range
-p	transport layer protocol (tcp, udp, or icmp)
-m	match a specific property (such as ‘state’)
--dport	destination port number (must be used with a protocol specified by the -p option)
--sport	source port number (must be used with a protocol specified by the -p option)
--state	connection state (NEW, ESTABLISHED, etc.)

An example rule using the above options is here:

```
# Allow inbound packets to TCP port 20 from subnet 192.168.1.0/24
iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 20 -j ACCEPT
```

Add rules to your /home/student/lab2/firewall.sh script that will allow outbound connection attempts on port 80 and the return traffic. The rules must perform stateful inspection and include the protocol and ports. Be very specific with your rules and include state, don’t just allow all.

Once you have edited and saved the firewall.sh file, apply at the command line as follows:

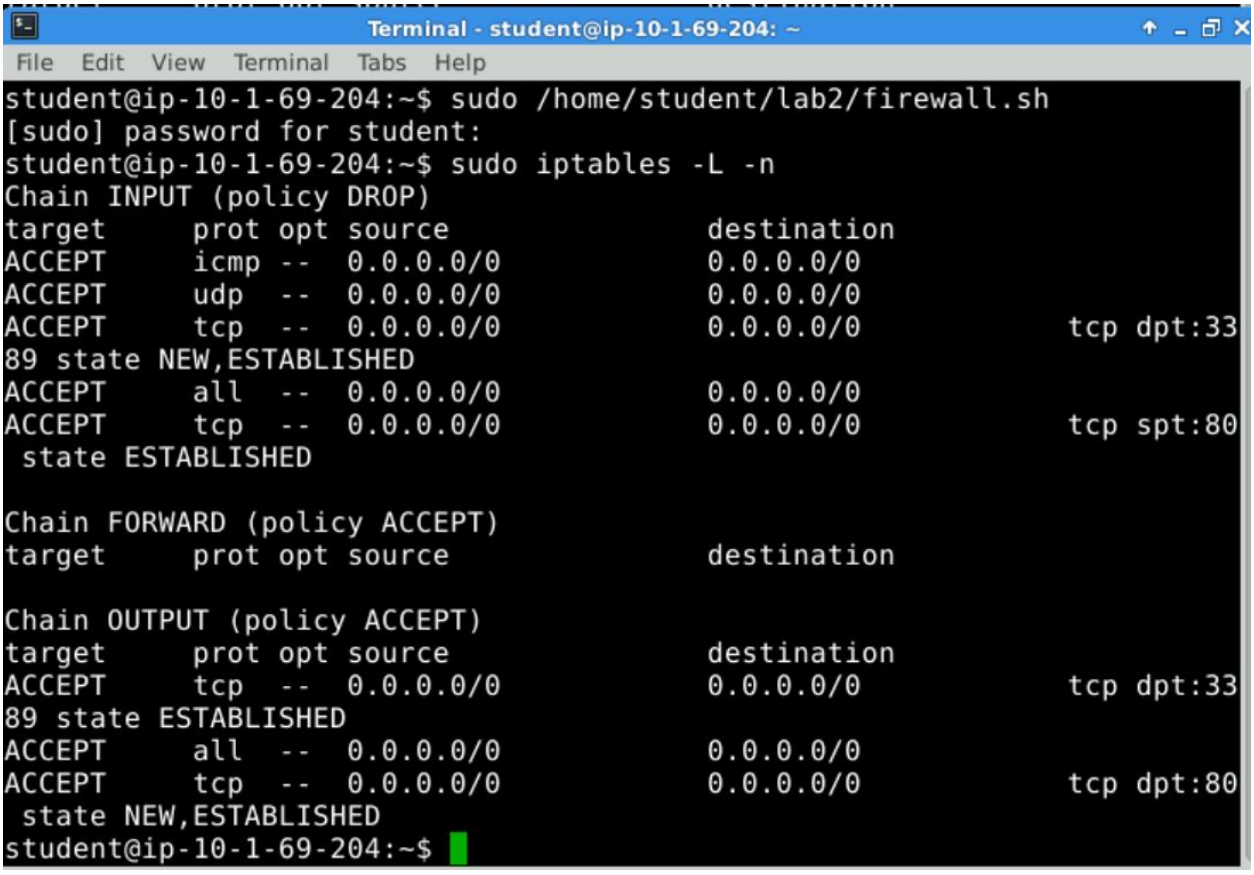
```
$ sudo /home/student/lab2/firewall.sh
```

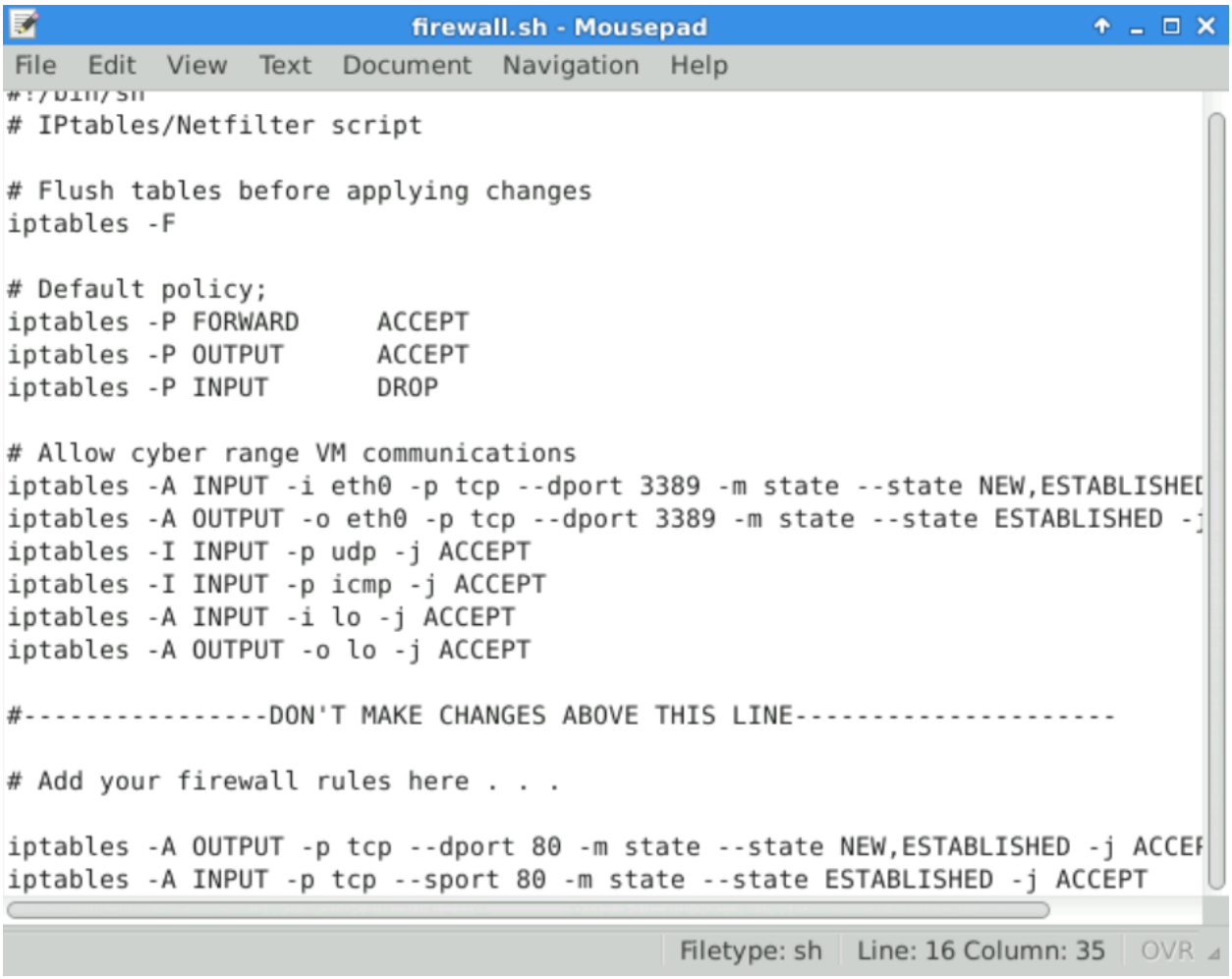
Perform the following command again to see that the new firewall rules are added:

```
$ sudo iptables -L -n
```

Question 6: **List the rule(s) that you added to the firewall.sh to allow outbound HTTP requests (port 80) and responses. (Show what you entered into the firewall.sh file in mousepad and provide a screenshot of the new firewall.sh) (1 point)**

The rules I added to firewall.sh to allow outbound HTTP traffic and responses are:
iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT





Task 2: Intrusion Detection

Your Virginia Cyber Range virtual machine has Snort software installed for intrusion detection. Instead of observing traffic from a network interface, we will use Snort to process packet capture files (.pcap files) from previously captured traffic.

Here is a great Snort reference from the Snort creator:
https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm

Before we run Snort against captured packets, we'll take a look at some snort rules (signatures) in the /etc/snort/rules directory. To do this, open a terminal window and change to the appropriate directory as follows.

```
$ cd /etc/snort/rules
$ ls                    ← this will list all the rule files
```

Examine the file **`shellcode.rules`** using the text editor of your choice (your Linux VM includes *vi* and *nano*, as well as a GUI text editor called *mousepad* as shown in the command below. You could also use the *cat* or *more* command).

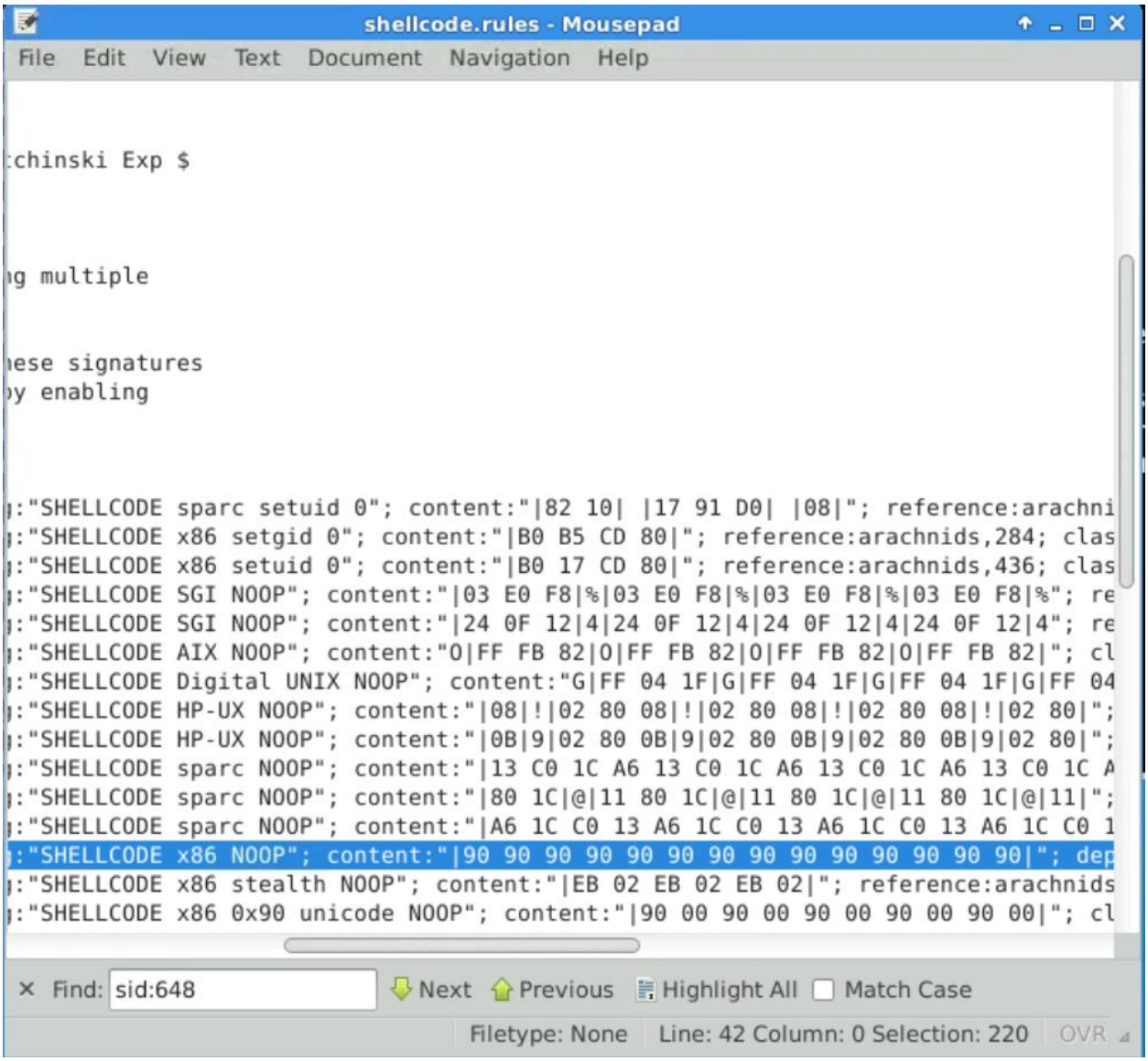
```
$ mousepad shellcode.rules &
```

Each rule has a unique Snort ID number (sid), which is included in the signature. In *shellcode.rules*, find **sid:648** amongst the rules in that file and answer the following questions.

Question 7: **What is the specific signature (content) that sid:648 tries to match and what would this indicate?(.5 point)**

The signature for sid:648 is the byte pattern
content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|";

This is a sequence of x86 0x90 bytes (NOP instructions), often used as a NOP sled in shellcode. Matching this pattern indicates possible x86 shellcode in the traffic, suggesting an attempted exploit or buffer-overflow attack.



Question 8: **What action is Snort supposed to take if the signature contained in sid:648 is matched? (.5 point)**

For sid:648, the rule starts with:
alert ip $EXTERNAL_NET any -> $HOME_NET $SHELLCODE_PORTS (... sid:648; ...)

So the action is alert — if that signature matches, Snort will generate an alert (log/report the event) indicating suspected shellcode, but it does not drop or block the packet itself.

Change directories to **/home/student/lab2** and run **snort** against the packet capture file called **theft.pcap** in that directory as shown here.

```
$ sudo snort -c /etc/snort/snort.conf -r theft.pcap
```

[When using ***sudo*** you may need to enter your student password: ***student***]

You will see Snort processing on your screen, let it complete and return on the command prompt $.

When Snort finishes processing, open a web browser on your Cyber Range virtual machine (on the menu bar at the bottom of the screen) It will automatically open the following URL: http://localhost/base.

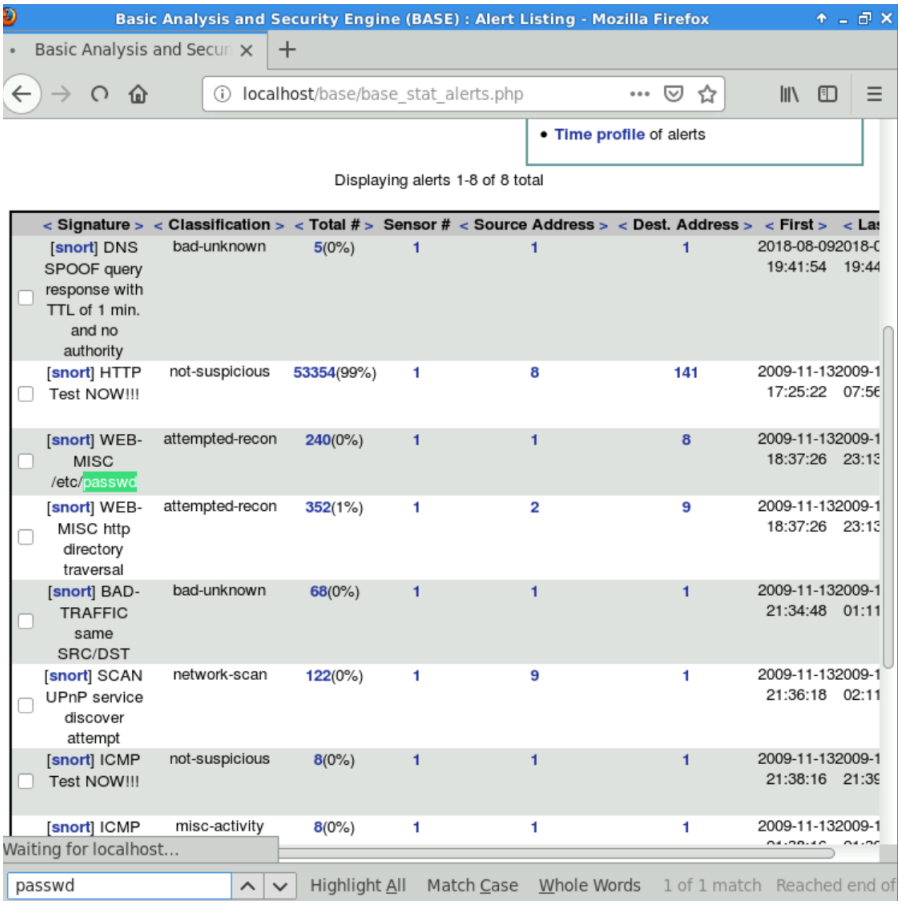
BASE is the Basic Analysis and Security Engine, which is installed on your virtual machine along with Snort. It allows you to view Snort alerts in a nice graphical format. Log in to BASE using the username: **john** and the password: **secret3**. The homepage should show the results of the scan we just processed using Snort, with a little over 27,000 Total Number of Alerts. [If you aren't seeing "Total Number of Alerts: 27081, It takes a few minutes for the back-end alert processing to complete, so you might have to refresh the page a few times.] NOTE: If you run the command with theft.pcap more than once you will have double or triple the alerts, **so only run it once**.

Review the unique alerts (you might have to do some filtering) and answer the following questions.

Question 9: **An attacker was trying to steal a specific, and very sensitive, file from the target system. What file were they after? Provide a screenshot and indicate where you found the answer. (.5 point)**

The attacker was trying to steal the /etc/passwd file.

I found this in BASE by going to the alert listing and looking at the Signature column, where one of the unique alerts is labeled [snort] WEB-MISC /etc/passwd, indicating access attempts to /etc/passwd.



Question 10: **There is another alert that shows the technique that the attacker was trying to use to steal the file. What is the technique? Provide a screenshot and indicate where you found the answer. (1 point)**

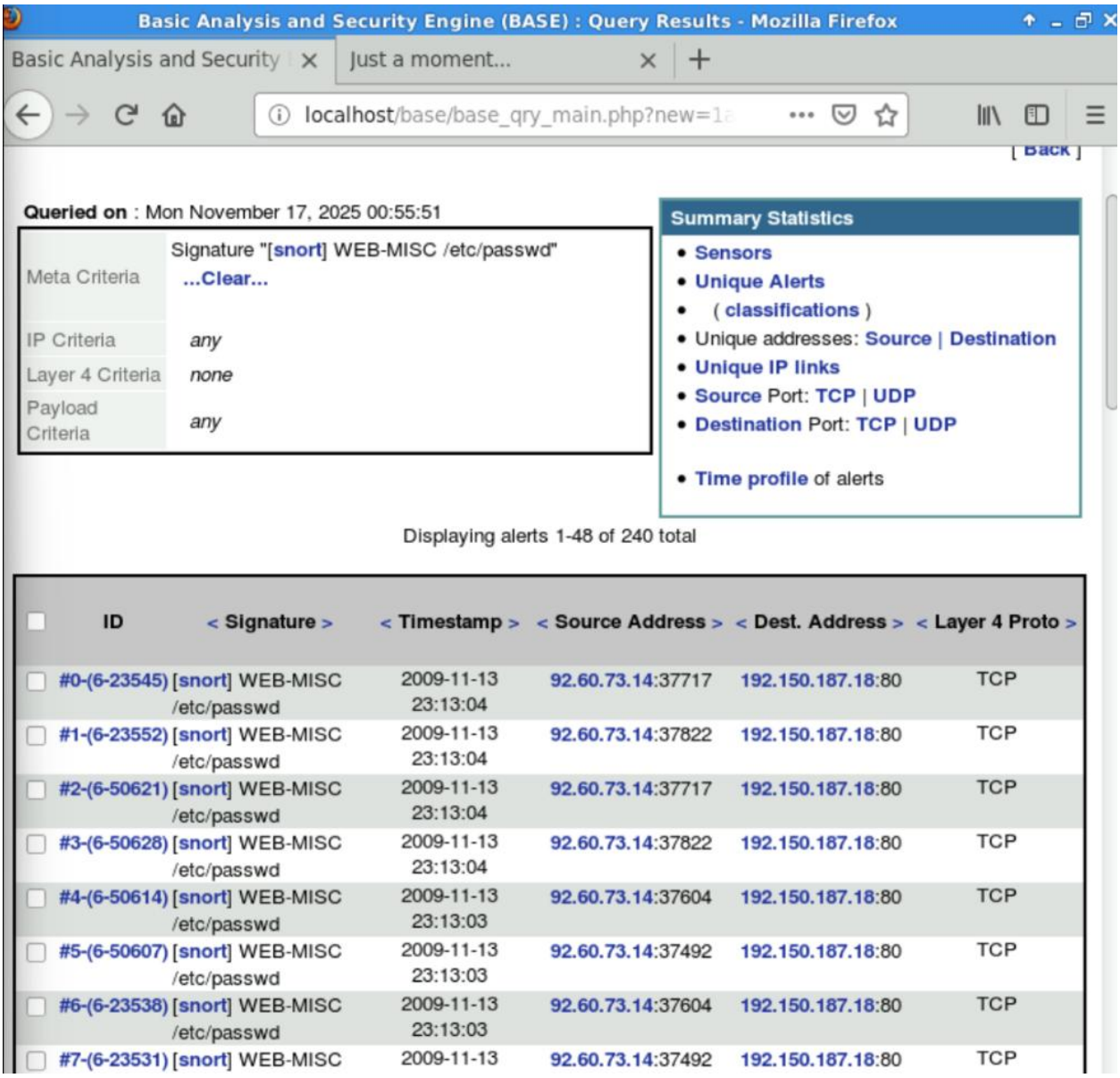
The attacker was using an HTTP directory traversal attack (using ../ style path traversal) to try to reach the /etc/passwd file.

In BASE, on the Unique Alerts page (the screenshot you sent), one of the signatures is [snort] WEB-MISC http directory traversal.

That alert indicates the technique: the attacker tried to move up directories via HTTP requests (e.g., ../../../../etc/passwd) to access files outside the web root.

Question 11: **What is the source IP address of the attacker that is trying to steal the file from the system? Provide a screenshot and indicate where you found the answer. (1 point)**

The source IP address of the attacker is 92.60.73.14.
I found this in BASE by clicking on the [snort] WEB-MISC /etc/passwd alert (Unique Alerts → Total # link). In the resulting alert list, the Source Address column shows 92.60.73.14 for the /etc/passwd requests (as seen in my screenshot).



By submitting this assignment you are digitally signing the honor code, "I pledge that I have neither given nor received help on this assignment".

END OF EXERCISE

References

- iptables man page: https://linux.die.net/man/8/iptables
- iptables tutorial: https://fideloper.com/iptables-tutorial
- Snort: https://www.snort.org/
- BASE: https://lwn.net/Articles/112548/